Outline

◆ Symbolic minimization

◆ Encoding problems:
  ▲ Input encoding
  ▲ Output encoding
  ▲ Mixed encoding
Symbolic minimization

◆ Minimize tables of *symbols* rather than binary tables
  ▲ Extension to bvi and mvi function minimization

◆ Applications:
  ▲ Simplification of tables of symbols
  ▲ Simplification of interconnected logic blocks
  ▲ Encoding of *finite-state machines*

◆ Problems:
  ▲ Input encoding
  ▲ Output encoding
  ▲ Mixed encoding
Example
(input encoding)

INSTRUCTION DECODER

ad-mode  op-code  control
## Example

<table>
<thead>
<tr>
<th>ad-mode</th>
<th>op-code</th>
<th>control</th>
</tr>
</thead>
<tbody>
<tr>
<td>INDEX</td>
<td>AND</td>
<td>CNTA</td>
</tr>
<tr>
<td>INDEX</td>
<td>OR</td>
<td>CNTA</td>
</tr>
<tr>
<td>INDEX</td>
<td>JMP</td>
<td>CNTA</td>
</tr>
<tr>
<td>INDEX</td>
<td>ADD</td>
<td>CNTA</td>
</tr>
<tr>
<td>DIR</td>
<td>AND</td>
<td>CNTB</td>
</tr>
<tr>
<td>DIR</td>
<td>OR</td>
<td>CNTB</td>
</tr>
<tr>
<td>DIR</td>
<td>JMP</td>
<td>CNTC</td>
</tr>
<tr>
<td>DIR</td>
<td>ADD</td>
<td>CNTC</td>
</tr>
<tr>
<td>IND</td>
<td>AND</td>
<td>CNTB</td>
</tr>
<tr>
<td>IND</td>
<td>OR</td>
<td>CNTD</td>
</tr>
<tr>
<td>IND</td>
<td>JMP</td>
<td>CNTD</td>
</tr>
<tr>
<td>IND</td>
<td>ADD</td>
<td>CNTC</td>
</tr>
</tbody>
</table>
Definitions

◆ Symbolic cover:
  ▲ List of symbolic implicants
  ▲ List of rows of a table

◆ Symbolic implicant:
  ▲ Conjunction of symbolic literals

◆ Symbolic literals:
  ▲ Simple: one symbol
  ▲ Compound: the disjunction of some symbols
Input encoding problem
Rationale

◆ Degrees of freedom in encoding the symbols

◆ Goal:
  ▲ Reduce size of the representation

◆ Approach:
  ▲ Encode to minimize number of rows
  ▲ Encode to minimize number of bits
Input encoding problem

- Represent each string by 1-hot codes
- Table with positional cube notation
- Minimize table with mvi minimizer
- Interpret minimized table:
  - Compound mvi-literals
  - Groups of symbols
**Example**

- **Encoded cover:**

  | 100 | 1000 | 1000 |
  | 100 | 0100 | 1000 |
  | 100 | 0010 | 1000 |
  | 100 | 0001 | 1000 |
  | 010 | 1000 | 0100 |
  | 010 | 0100 | 0100 |
  | 010 | 0010 | 0010 |
  | 010 | 0001 | 0010 |
  | 001 | 1000 | 0100 |
  | 001 | 0100 | 0001 |
  | 001 | 0010 | 0001 |
  | 001 | 0001 | 0010 |

- **Minimum cover:**

  | 100 | 1111 | 1000 |
  | 010 | 1100 | 0100 |
  | 001 | 1000 | 0100 |
  | 010 | 0011 | 0010 |
  | 001 | 0010 | 0010 |
  | 001 | 0110 | 0001 |
Example

◆ Minimum symbolic cover:

<table>
<thead>
<tr>
<th>INDEX</th>
<th>AND, OR, JMP, ADD</th>
<th>CNTA</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIR</td>
<td>AND, OR</td>
<td>CNTB</td>
</tr>
<tr>
<td>IND</td>
<td>AND</td>
<td>CNTB</td>
</tr>
<tr>
<td>DIR</td>
<td>JMP, ADD</td>
<td>CNTC</td>
</tr>
<tr>
<td>IND</td>
<td>ADD</td>
<td>CNTC</td>
</tr>
<tr>
<td>IND</td>
<td>OR, JMP</td>
<td>CNTD</td>
</tr>
</tbody>
</table>

◆ Examples of:

▲ Simple literal: AND

▲ Compound literal: AND, OR
Input encoding problem

- Transform minimum symbolic cover into minimum bv-cover
- Map symbolic implicants into bv implicants (one to one)
- Compound literals:
  - Encode corresponding symbols so that their supercube does not include other symbol codes
- Replace encoded literals into cover
Example

◆ Compound literals:

▲ AND, OR, JMP, ADD
▲ AND, OR
▲ JMP, ADD
▲ OR, JMP
Example

◆ Valid codes:

- **AND** 00
- **OR** 01
- **JMP** 11
- **ADD** 10

◆ Replacement in cover:

- **Final cover**

  - 1111 → **00**
  - 1100 → **01**
  - 1000 → **00**
  - 0011 → **10**
  - 0010 → **11**
  - 0110 → **11**

- (a)

  - 00  01  00  10  01  10  00  11  00  11  01  11  00  00  00  11  00  00  11  00
Input encoding algorithms

Problem specification:

- Constraint matrix $A$:
  - $a_{ij} = 1$ iff symbol $j$ belongs to literal $i$

Solution sought for:

- Encoding matrix $E$:
  - As many rows as the symbols
  - Encoding length $n_b$
Example

◆ Constraint matrix:

\[ A = \begin{bmatrix}
1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 \\
1 & 1 & 1 & 0
\end{bmatrix} \]

◆ Encoding matrix:

\[ E = \begin{bmatrix}
0 & 0 \\
0 & 1 \\
1 & 1 \\
1 & 0
\end{bmatrix} \]

Compound literals:

▲ AND, OR, JMP, ADD = *
▲ AND, OR
▲ JMP, ADD
▲ OR, JMP
Input encoding problem

Given constraint matrix $A$:

- Find encoding matrix $E$ satisfying all input encoding constraints (due to compound literals)
- With minimum number of columns (bits)
Dichotomy theory

◆ Dichotomy:
  ▲ Two sets \((L, R)\)
  ▲ Bipartition of a subset of the symbol set

◆ Encoding:
  ▲ Set of columns of \(E\)
  ▲ Set of bipartitions of symbols set

◆ Rationale:
  ▲ Each row of the constraint matrix implies some choice on the codes
Dichotomies

- Dichotomy associated with row $a^T$ of $A$:
  - A set pair $(L, R)$
    - $L$ has the symbols with the 1s in $a^T$
    - $R$ has the symbols with the Os in $a^T$

- Seed dichotomy associated with row $a^T$ of $A$:
  - A set pair $(L, R)$
    - $L$ has the symbols with the 1s in $a^T$
    - $R$ has one symbol with the O in $a^T$
Example

◆ Dichotomy associated with constraint $a^T = 1100$:
  ▲ $\{\text{AND, OR}\}; \{\text{JMP, ADD}\}$

◆ The corresponding seed dichotomies are:
  ▲ $\{\text{AND, OR}\}; \{\text{JMP}\}$
  ▲ $\{\text{AND, OR}\}; \{\text{ADD}\}$
Definitions

◆ Compatibility:

△ \((L_1; R_1)\) and \((L_2; R_2)\) are compatible if:

\[ L_1 \cap R_2 = \emptyset \text{ and } R_1 \cap L_2 = \emptyset \]

◆ Covering:

△ Dichotomy \((L_1; R_1)\) covers \((L_2; R_2)\) if:

\[ L_1 \geq L_2 \text{ and } R_1 \geq R_2 \]

◆ Prime dichotomy:

△ Dichotomy that is not covered by any compatible dichotomy of a given set
Extended definitions

◆ Compatibility:

\(\Delta (L_1; R_1) \text{ and } (L_2; R_2)\) are compatible if:

\(\nabla L_1 \cap R_2 = \emptyset\) and \(R_1 \cap L_2 = \emptyset\)

or

\(\nabla L_1 \cap L_2 = \emptyset\) and \(R_1 \cap R_2 = \emptyset\)

◆ Covering:

\(\Delta\) Dichotomy \((L_1; R_1)\) covers \((L_2; R_2)\) if:

\(\nabla L_1 \geq L_2\) and \(R_1 \geq R_2\)

or

\(\nabla L_1 \geq R_2\) and \(R_1 \geq L_2\)

◆ Prime dichotomy:

\(\Delta\) Dichotomy that is not covered by any compatible dichotomy of a given set
Exact input encoding

- Compute all prime dichotomies
- Form a prime/seed table
- Find minimum cover of seeds by primes
Example

◆ Seed dichotomies:

<table>
<thead>
<tr>
<th>s</th>
<th>{AND, OR} ;  {JMP}</th>
</tr>
</thead>
<tbody>
<tr>
<td>s_1</td>
<td>{AND, OR} ;  {ADD}</td>
</tr>
<tr>
<td>s_2</td>
<td>{JMP, ADD} ;  {AND}</td>
</tr>
<tr>
<td>s_3</td>
<td>{JMP, ADD} ;  {OR}</td>
</tr>
<tr>
<td>s_4</td>
<td>{OR, JMP} ;  {AND}</td>
</tr>
<tr>
<td>s_5</td>
<td>{OR, JMP} ;  {ADD}</td>
</tr>
<tr>
<td>s_6</td>
<td>{OR, JMP} ;  {ADD}</td>
</tr>
</tbody>
</table>

◆ Primes dichotomies:

<table>
<thead>
<tr>
<th>p</th>
<th>{AND, OR} ;  {JMP, ADD}</th>
</tr>
</thead>
<tbody>
<tr>
<td>p_1</td>
<td>{OR, JMP} ;  {AND, ADD}</td>
</tr>
<tr>
<td>p_2</td>
<td>{OR, JMP, ADD} ;  {AND}</td>
</tr>
<tr>
<td>p_3</td>
<td>{AND, OR, JMP} ;  {ADD}</td>
</tr>
<tr>
<td>p_4</td>
<td>{AND, OR, JMP} ;  {ADD}</td>
</tr>
</tbody>
</table>
**Example**

- **Table:**

<table>
<thead>
<tr>
<th></th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$s_4$</th>
<th>$s_5$</th>
<th>$s_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_1$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$p_2$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$p_3$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$p_4$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

- **Minimum cover:** $p_1$ and $p_2$

- **Encoding:**

$$E = \begin{pmatrix}
1 & 0 \\
1 & 1 \\
0 & 1 \\
0 & 0
\end{pmatrix}$$

$p_1$ ({$\text{AND, OR}$}; {$\text{JMP, ADD}$})

$p_2$ ({$\text{OR, JMP}$}; {$\text{AND, ADD}$})
Remarks

- Satisfying all encoding constraints may require more than the minimum number of bits $\log_2 n$ for $n$ symbols.
- 1-hot encoding satisfies all possible constraint sets, but $n$ bits are needed.
- Trade-off is possible between coding length and constraint satisfaction, that then relates to minimality of the cover.
Heuristic encoding

◆ Determine dichotomies of rows of $A$

◆ Column-based encoding:
  ▲ Construct $E$ column by column

◆ Iterate:
  ▲ Determine maximum compatible set
  ▲ Find a compatible encoding
  ▲ Use it as column of $E$
Example

◆ Dichotomies

\[
\begin{array}{c|cc}
\text{d}_1 & \{\text{AND, OR}\} & \{\text{JMP, ADD}\} \\
\text{d}_2 & \{\text{JMP, ADD}\} & \{\text{AND, OR}\} \\
\text{d}_3 & \{\text{OR, JMP}\} & \{\text{AND, ADD}\} \\
\end{array}
\]

◆ First two dichotomies are compatible

◆ Encoding column \([1100]^T\) satisfies \(d_1, d_2\)

◆ Need to satisfy \(d_3\)

◆ Second encoding column \([0110]^T\)
Output and mixed encoding

Output encoding:

- Determine encoding of output symbols

Mixed encoding:

- Determine both input and output encoding

Examples

- Interconnected circuits
- Circuits with feedback
Example

INSTRUCTION DECODER

ad-mode  op-code  control
Symbolic minimization

◆ Extension to mvi-minimization

◆ Accounts for:
  ▲ Covering relations
  ▲ Disjunctive relations

◆ Exact and heuristic minimizers
Example

◆ Minimum symbolic cover computed before:

<table>
<thead>
<tr>
<th>INDEX</th>
<th>AND, OR, JMP, ADD</th>
<th>CNTA</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIR</td>
<td>AND, OR</td>
<td>CNTB</td>
</tr>
<tr>
<td>IND</td>
<td>AND</td>
<td>CNTB</td>
</tr>
<tr>
<td>DIR</td>
<td>JMP, ADD</td>
<td>CNTC</td>
</tr>
<tr>
<td>IND</td>
<td>ADD</td>
<td>CNTC</td>
</tr>
<tr>
<td>IND</td>
<td>OR, JMP</td>
<td>CNTD</td>
</tr>
</tbody>
</table>

◆ Can we use fewer implicants?
◆ Can we merge implicants?
## Example

<table>
<thead>
<tr>
<th>INDEX</th>
<th>AND, OR, JMP, ADD</th>
<th>CNTA</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIR</td>
<td>AND, OR</td>
<td>CNTB</td>
</tr>
<tr>
<td>IND</td>
<td>AND</td>
<td>CNTB</td>
</tr>
<tr>
<td>DIR</td>
<td>JMP, ADD</td>
<td>CNTC</td>
</tr>
<tr>
<td>IND</td>
<td>ADD</td>
<td>CNTC</td>
</tr>
<tr>
<td>IND</td>
<td>OR, JMP</td>
<td>CNTD</td>
</tr>
</tbody>
</table>

**Merging implicants:**

<table>
<thead>
<tr>
<th>INDEX</th>
<th>AND, OR, JMP, ADD</th>
<th>CNTA</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIR, IND</td>
<td>AND, OR</td>
<td>CNTB</td>
</tr>
<tr>
<td>DIR, IND</td>
<td>JMP, ADD</td>
<td>CNTC</td>
</tr>
<tr>
<td>IND</td>
<td>OR, JMP</td>
<td>CNTD</td>
</tr>
</tbody>
</table>

**Conflict:**

- Conditions for CNTD overlaps conditions for CNTB and CNTC
- Make sure CNTD overrides the others → bitwise cover

(c) Giovanni De Micheli
Example
covering relations

◆ Assume the code of \textit{CNTD} covers the codes of \textit{CNTB} and \textit{CNTC}:

\begin{center}
\begin{tabular}{ccc}
100 & 1111 & CNTA \\
001 & 1100 & CNTB \\
011 & 0011 & CNTC \\
001 & 0110 & CNTD \\
\end{tabular}
\end{center}

◆ Possible codes:

\begin{itemize}
\item \textcolor{red}{\textit{CNTA} = 00, \textit{CNTB} = 01, \textit{CNTC} = 10 and \textit{CNTD} = 11}
\end{itemize}
Output encoding algorithms

- Often solved in conjunction with input encoding

- Exact algorithms:
  - Prime dichotomies compatible with output constraints
  - Asymmetry: L not interchangeable with R
  - Construct prime / seed table
  - Solve covering problem

- Heuristic algorithms:
  - Construct $E$ column by column
Example

◆ Input constraint matrix of second stage:

\[ A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \]

◆ Output constraint matrix of first stage:

\[ B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} \]

◆ Assume the code of \textit{CTND} covers the codes of \textit{CTNB} and \textit{CTNC}
Example

Seed dichotomies associated with A:

<table>
<thead>
<tr>
<th>$s_1$</th>
<th>${\text{CNTA, CNTB}}$</th>
<th>${\text{CNTC}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_2$</td>
<td>${\text{CNTA, CNTB}}$</td>
<td>${\text{CNTD}}$</td>
</tr>
<tr>
<td>$s_3$</td>
<td>${\text{CNTC}}$</td>
<td>${\text{CNTA, CNTB}}$</td>
</tr>
<tr>
<td>$s_4$</td>
<td>${\text{CNTD}}$</td>
<td>${\text{CNTA, CNTB}}$</td>
</tr>
<tr>
<td>$s_5$</td>
<td>${\text{CNTB, CNTD}}$</td>
<td>${\text{CNTA}}$</td>
</tr>
<tr>
<td>$s_6$</td>
<td>${\text{CNTB, CNTD}}$</td>
<td>${\text{CNTC}}$</td>
</tr>
<tr>
<td>$s_7$</td>
<td>${\text{CNTA}}$</td>
<td>${\text{CNTB, CNTD}}$</td>
</tr>
<tr>
<td>$s_8$</td>
<td>${\text{CNTC}}$</td>
<td>${\text{CNTB, CNTD}}$</td>
</tr>
</tbody>
</table>

Seed dichotomies $s_2$ and $s_8$ are not compatible with B
Example (2)

◆ Prime dichotomies compatible with B:

\[
\begin{align*}
\rho_1 & : \{\text{CNTC, CNTD}\} ; \{\text{CNTA, CNTB}\} \\
\rho_2 & : \{\text{CNTB, CNTD}\} ; \{\text{CNTA, CNTC}\} \\
\rho_3 & : \{\text{CNTA, CNTB, CNTD}\} ; \{\text{CNTC}\}
\end{align*}
\]

◆ Cover: \(\rho_1\) and \(\rho_2\)

◆ Encoding matrix

\[
E = \begin{pmatrix}
0 & 0 \\
0 & 1 \\
1 & 0 \\
1 & 1
\end{pmatrix}
\]

AND   OR   JMP   ADD

(c) Giovanni De Micheli
State encoding of finite-state machines

- Given a state table of a finite-state machine:
  - With symbols representing
    - present-states
    - next-states

- Find a consistent encoding of the states:
  - That minimizes the size of the cover
  - With minimum number of bits
Example

![Diagram of a combinational circuit with primary inputs, state, and primary outputs. The table below shows the input, P-state, N-state, and output values.]

<table>
<thead>
<tr>
<th>INPUT</th>
<th>P-STATE</th>
<th>N-STATE</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>s₁</td>
<td>s₃</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>s₁</td>
<td>s₃</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>s₂</td>
<td>s₃</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>s₂</td>
<td>s₁</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>s₃</td>
<td>s₅</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>s₃</td>
<td>s₄</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>s₄</td>
<td>s₂</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>s₄</td>
<td>s₅</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>s₅</td>
<td>s₂</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>s₅</td>
<td>s₅</td>
<td>0</td>
</tr>
</tbody>
</table>
Example

◆ Minimum symbolic cover:

\[
\begin{align*}
&* & s_1 & s_2 & s_4 & s_3 & 0 \\
&1 & s_2 & & s_1 & 1 \\
&0 & s_4 & s_5 & & s_2 & 1 \\
&1 & & s_3 & & s_4 & 1 \\
\end{align*}
\]

◆ Covering constraints:

\[s_1 \text{ and } s_2 \text{ cover } s_3\]

\[s_5 \text{ is covered by all other states}\]

◆ Constraint and encoding matrices:

\[
A = \begin{bmatrix}
1 & 1 & 0 & 1 & 0 \\
1 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 1
\end{bmatrix}
\]

\[
B = \begin{bmatrix}
0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

\[
E = \begin{bmatrix}
1 & 1 & 1 \\
1 & 0 & 1 \\
0 & 0 & 1 \\
1 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix}
\]
Example

- Encoding matrix (one row per state):

\[
E = \begin{bmatrix}
1 & 1 & 1 \\
1 & 0 & 1 \\
0 & 0 & 1 \\
1 & 0 & 0 \\
0 & 0 & 0 \\
\end{bmatrix}
\]

- Encoded cover of combinational component:

<table>
<thead>
<tr>
<th></th>
<th>1**</th>
<th>001</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>101</td>
<td>111</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>*00</td>
<td>101</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>001</td>
<td>100</td>
<td>1</td>
</tr>
</tbody>
</table>
Summary

◆ Symbolic minimization:
  ▲ Reduce size of tabular representations where symbols in table can be encoded

◆ Requires solving encoding problems:
  ▲ Find minimum-length encoding that is valid for a minimum symbolic cover

◆ Applicable to optimizing:
  ▲ Interconnected combinational blocks
  ▲ Combinational part of finite-state machines